Mechanism for 3-way Feature Interactions Occurrence and A Detection System based on The Mechanism

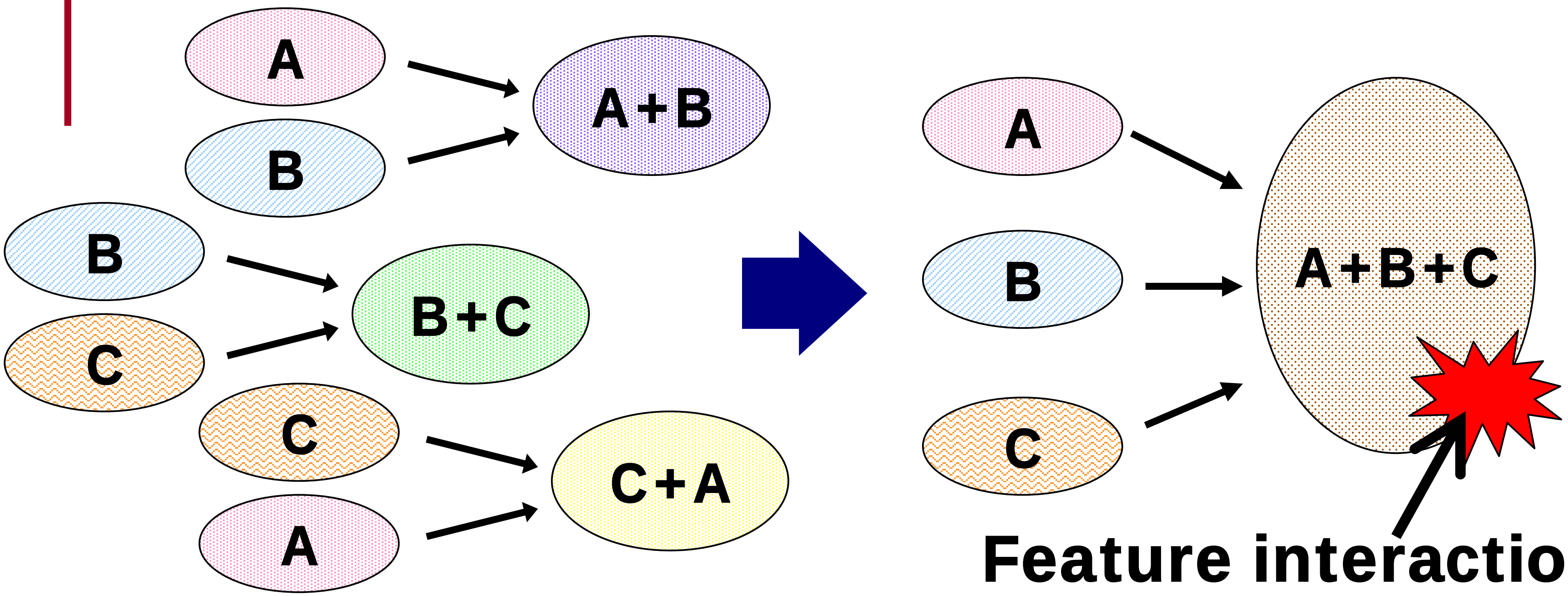
Shizuko Kawauchi, Tadashi Ohta
Soka University

Contents

- 3-way feature interaction
- Mechanism for 3-way interaction occurrence
- Detection algorithm for 3-way interactions
- Problem in implementing the detection system
- Discussion and Conclusion

3-way feature interaction

Feature interaction which does not occur between two services but occurs among three services, is called a 3-way interaction.

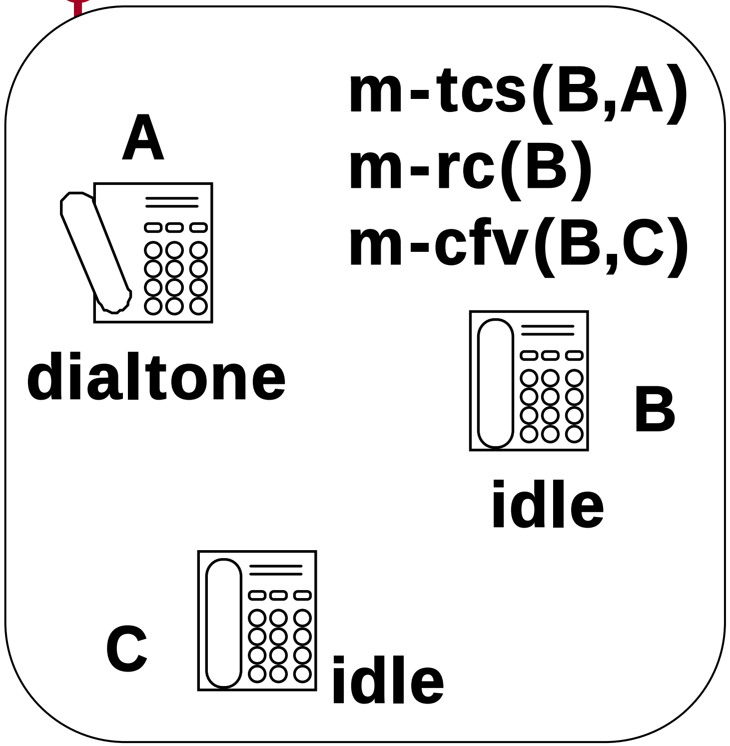


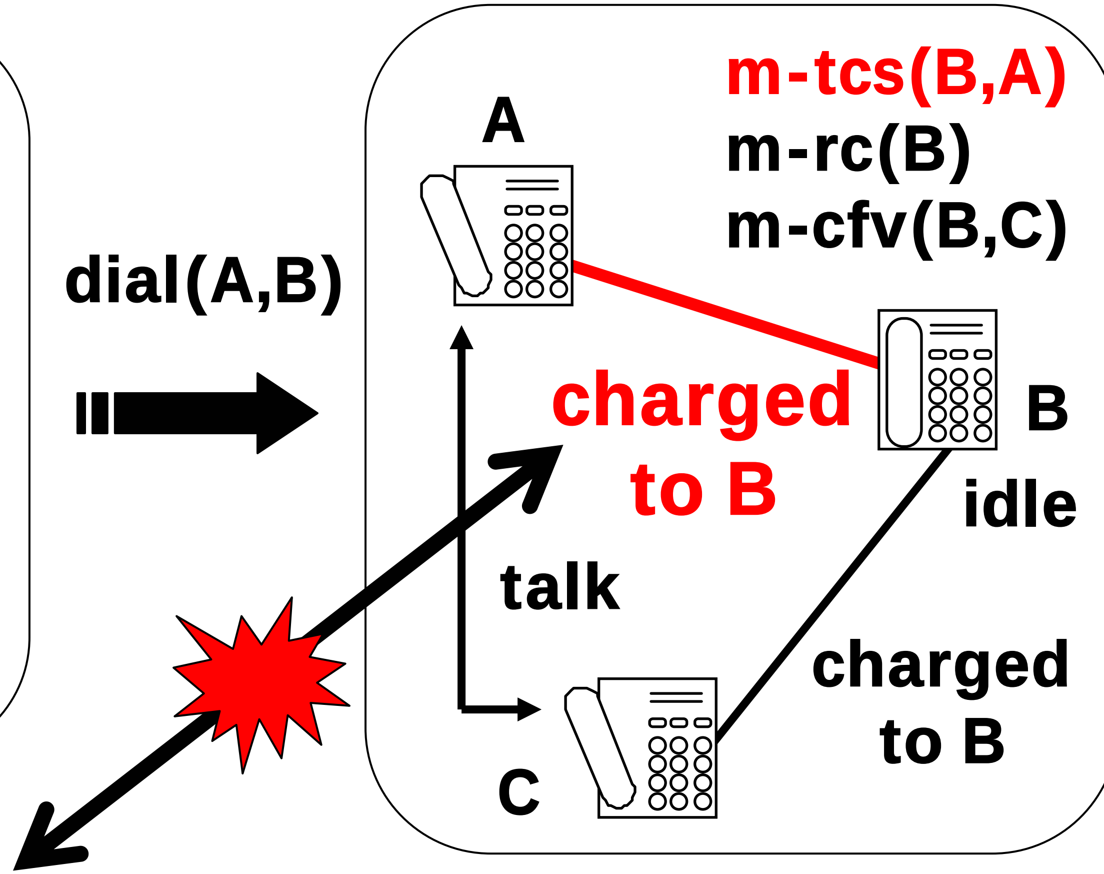
Feature interaction α does not occur.

Feature interaction α occurs.

⇨ 3-way interation

Example of 3-way interaction



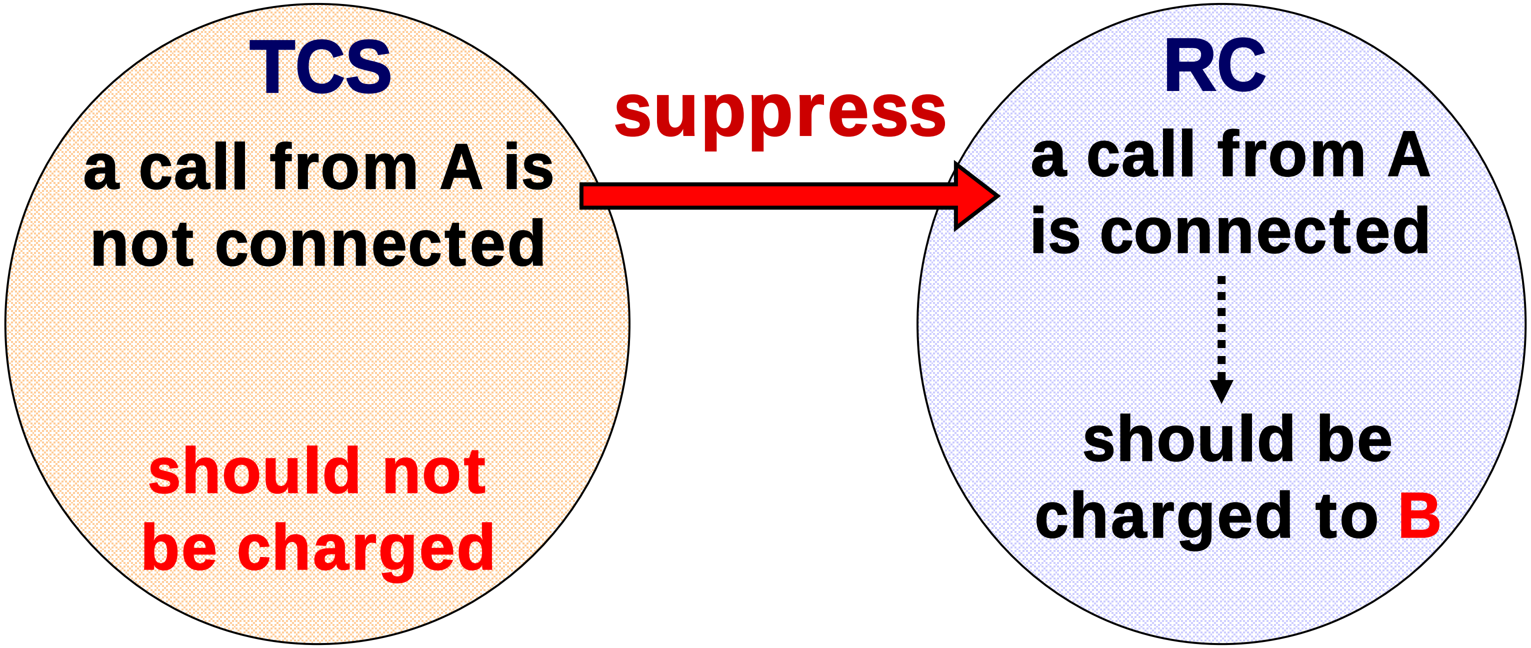


TCS:
The link between terminal A and B should not be charged.

The bill for a link between terminal A and B is feature interaction

Occurrence process(1)

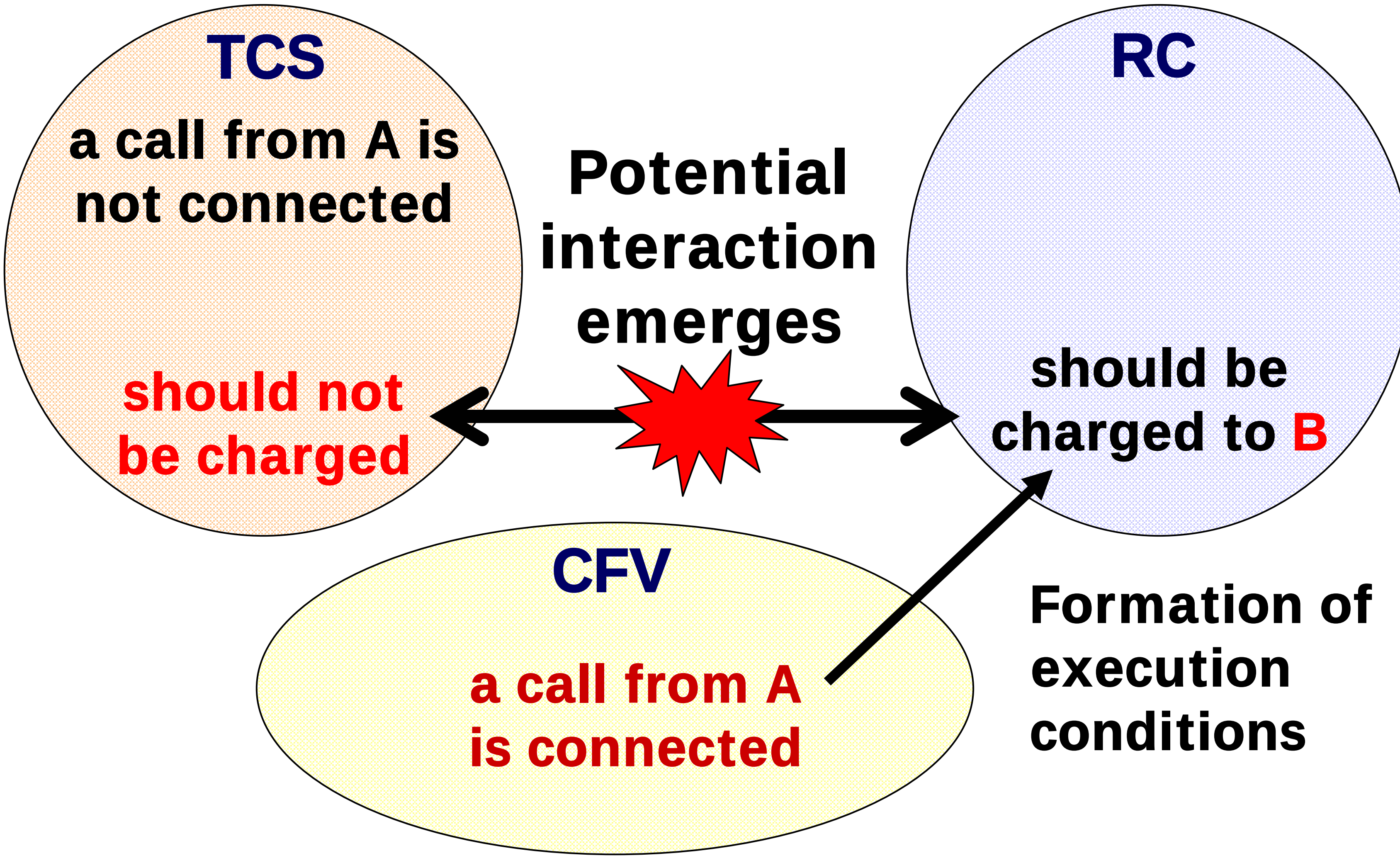
The contradictory specification between two service specifications is called a potential interaction.



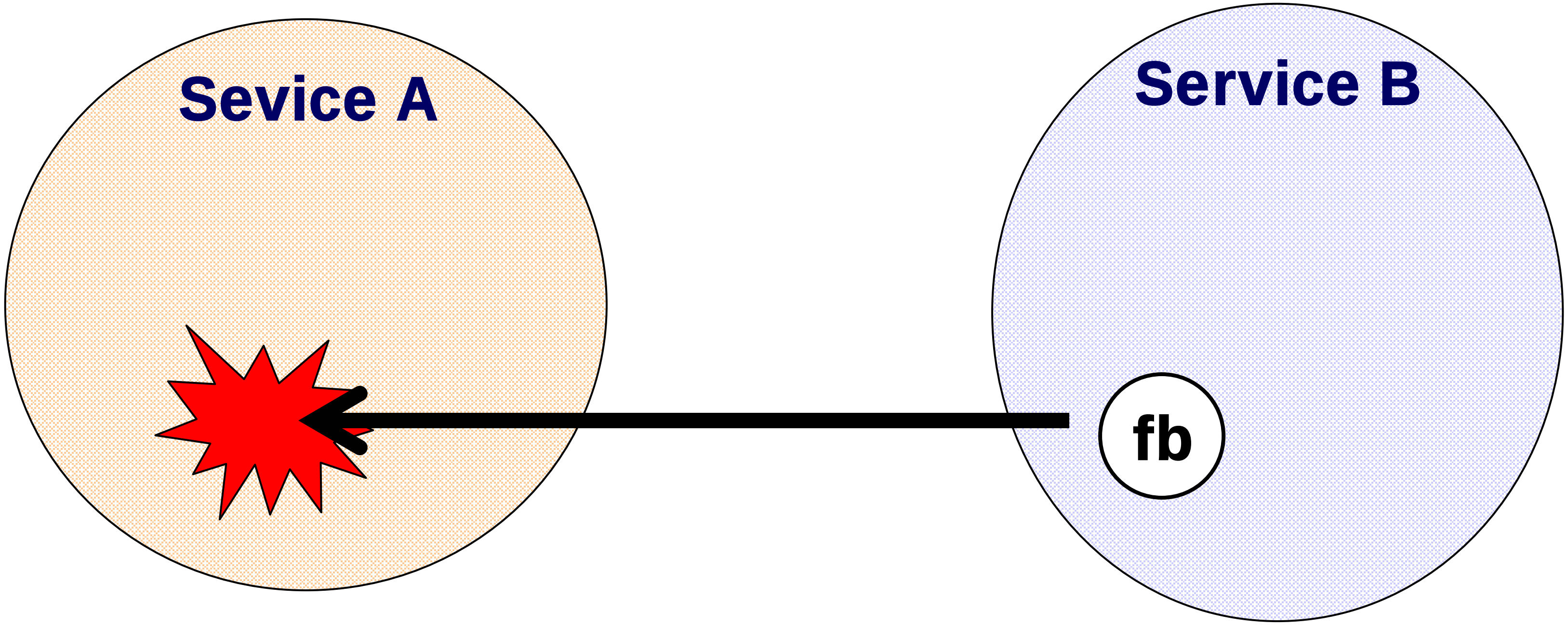
This potential interaction does not emerge.

Occurrence process(2)

Ex) TCS(B,A) & RC(B) & CFV(B,C)

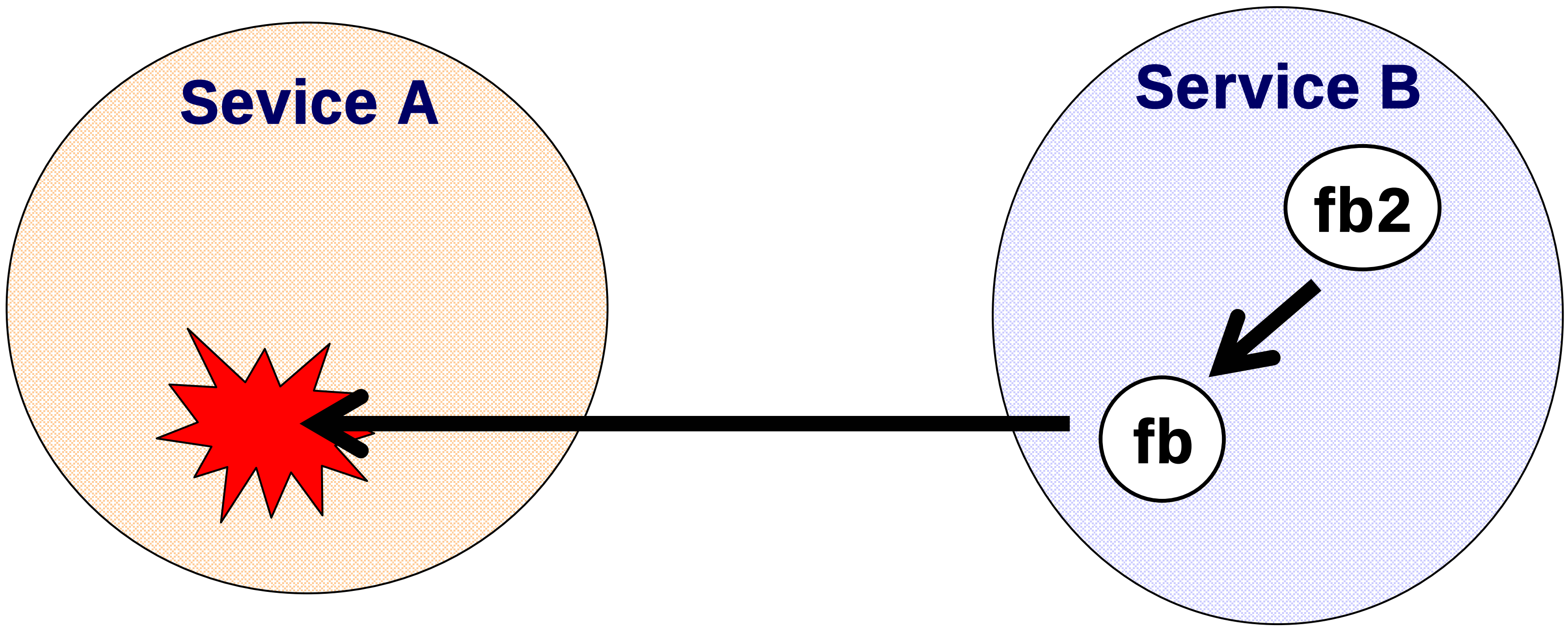


Mechanism(1)



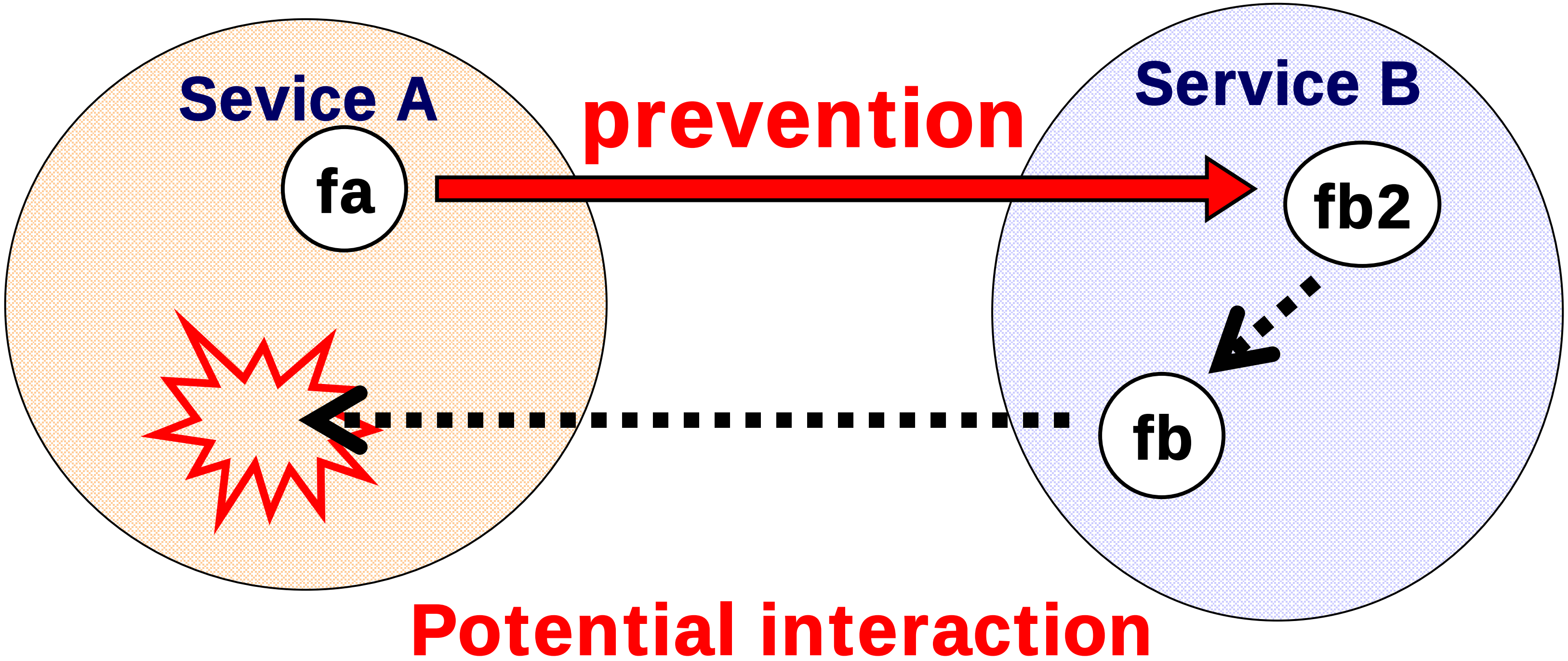
fb: Contradicts the specification of service A

Mechanism(2)



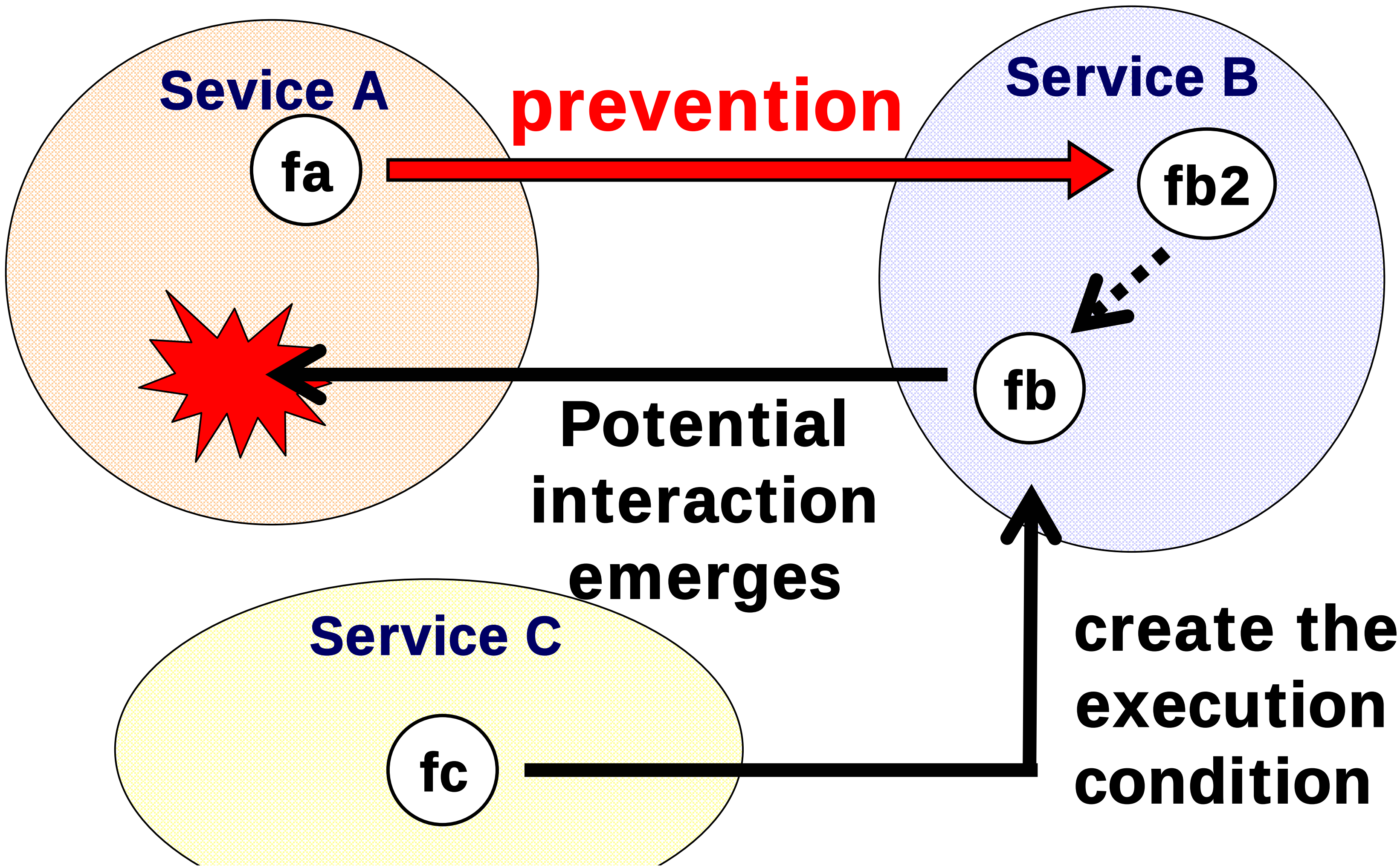
fb2: Creates the execution condition for fb

Mechanism(3)



fa: Prevents fb2 from being executed

Mechanism(4)



fc: Creates the execution condition for fb

Service specification ～STR

- STR(State Transition Rule) is a rule type language to define conditions for state transitions.
- Service specifications can be represented as a set of rules.

idle(x),dialtone(y)	dial(x,y)	:	Calling(x,y)

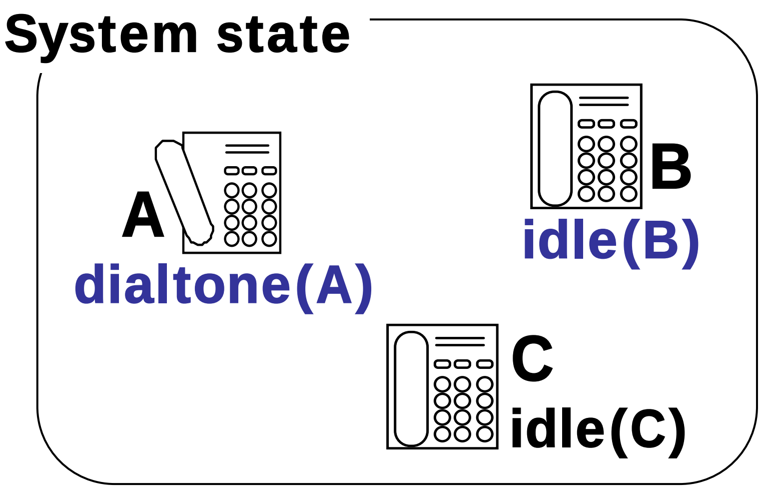
Pre-conditon	an execution condition of a rule
event	trigger for a state transition
Post-conditon	a system state condition after the state transition

All arguments in primitives are described as variables.

Application conditions for rules

A rule whose **Pre-condition exists in the system state** is selected and applied.

Ex) dialtone(x),idle(y) dial(x,y): Calling(x,y)



This rule is applied since Pre-condition of this rule exists in the system state.

If more than one rule are applicable, **the rule whose Pre-condition includes Pre-conditions of any other rules is applied**.

Detection algorithm

step1 Selection of a rule, rb which has feature with generating a potential interaction

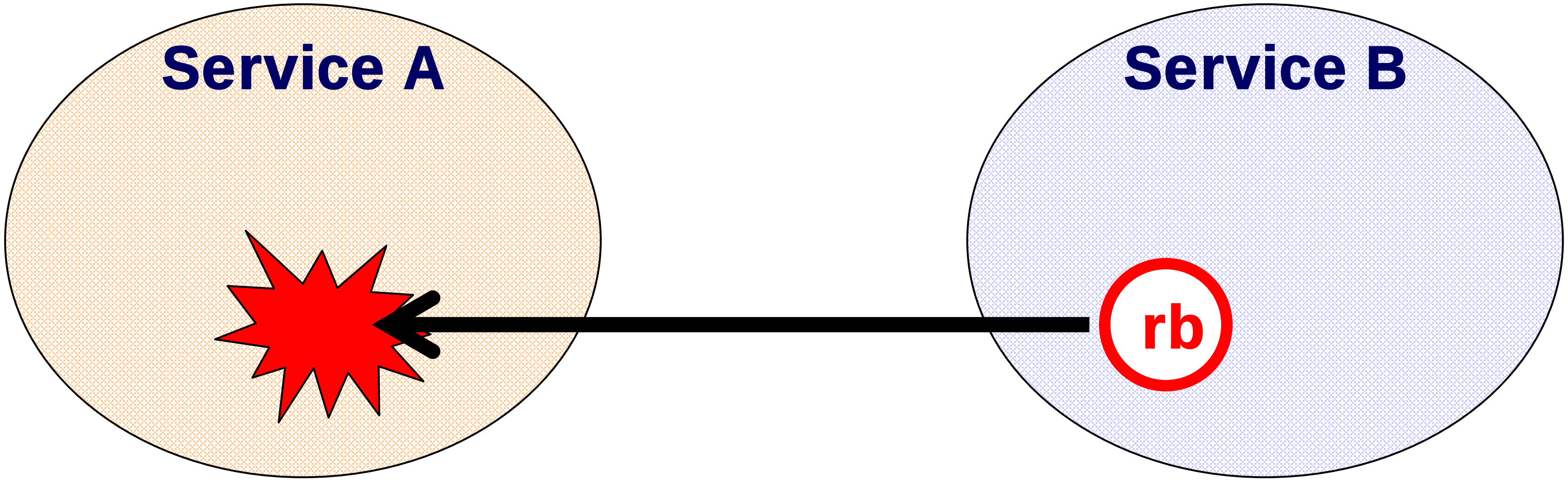
step2 Selection of a rule, rb2 which has feature with creating an execution condition of rb

step3 Selection of a rule, ra which be applied in precedence over rb2

step4 Selection of a rule, rc which has feature with creating an execution condition of rb

Step1 Selection of rb

Select rb which causes a state contradicting with specifications of service A.

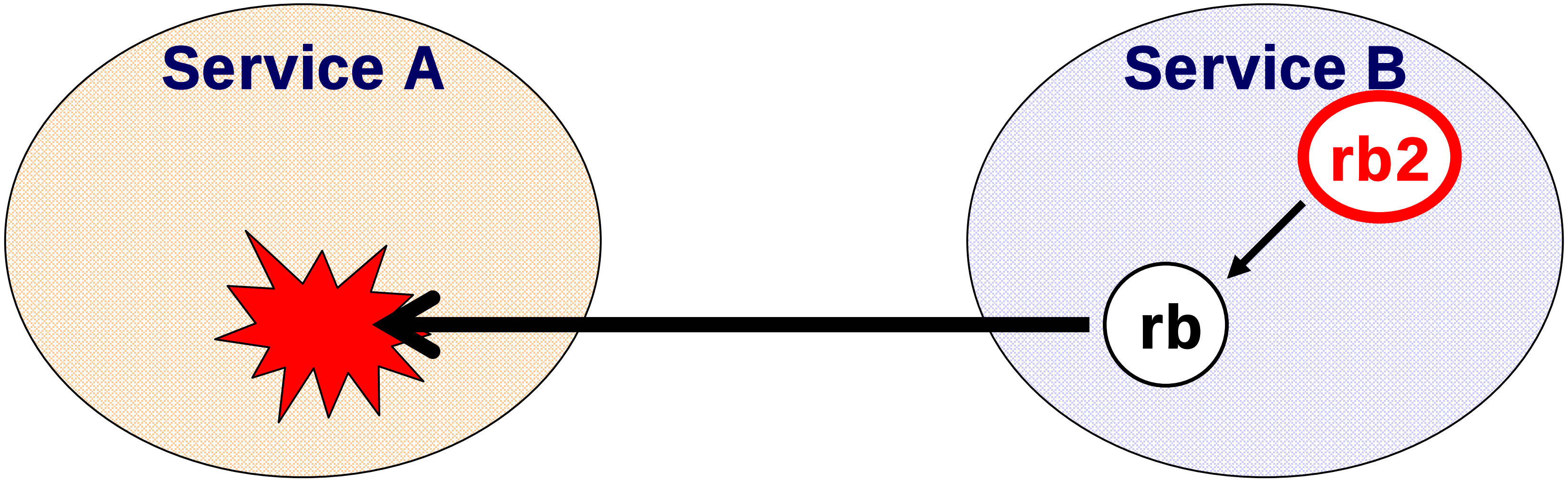


the Post-condition of rb ⊇

states which contradict with specifications of service A

Step2 Selection of rb2

Select rb2 which creates execution conditions for rb.

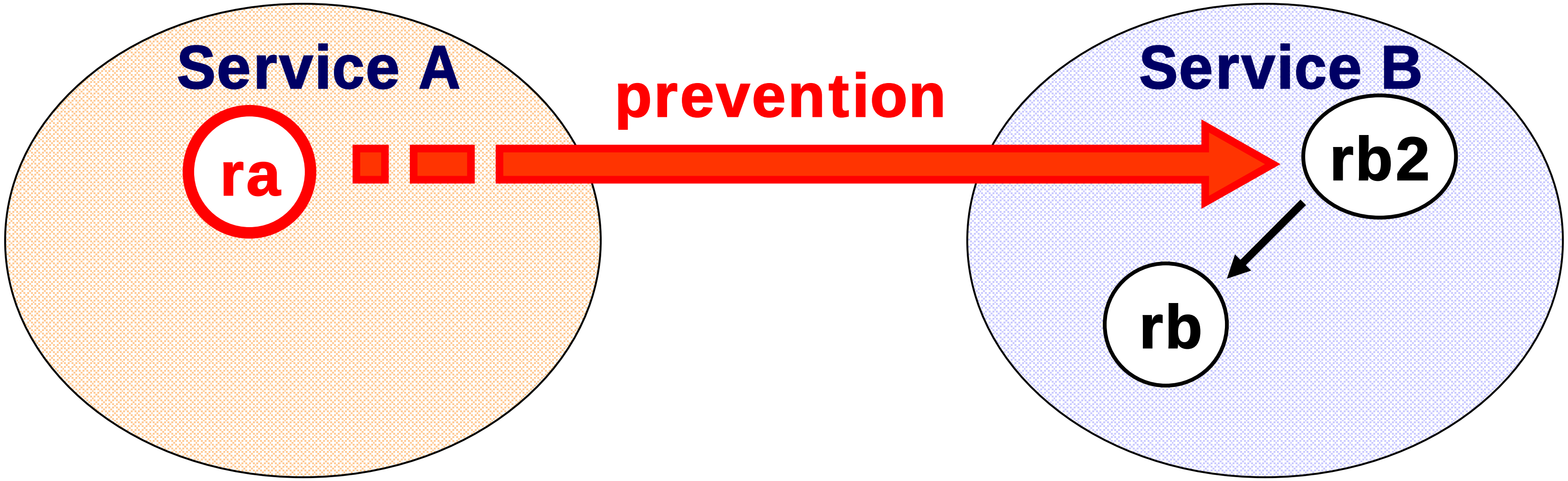


the Post-condition of rb2

$\supseteq$ the Pre-condition of rb

Step3 Selection of ra

Select ra which can be applied in precedence over rb2 and does not create execution conditions for rb.



the Pre-condition of ra

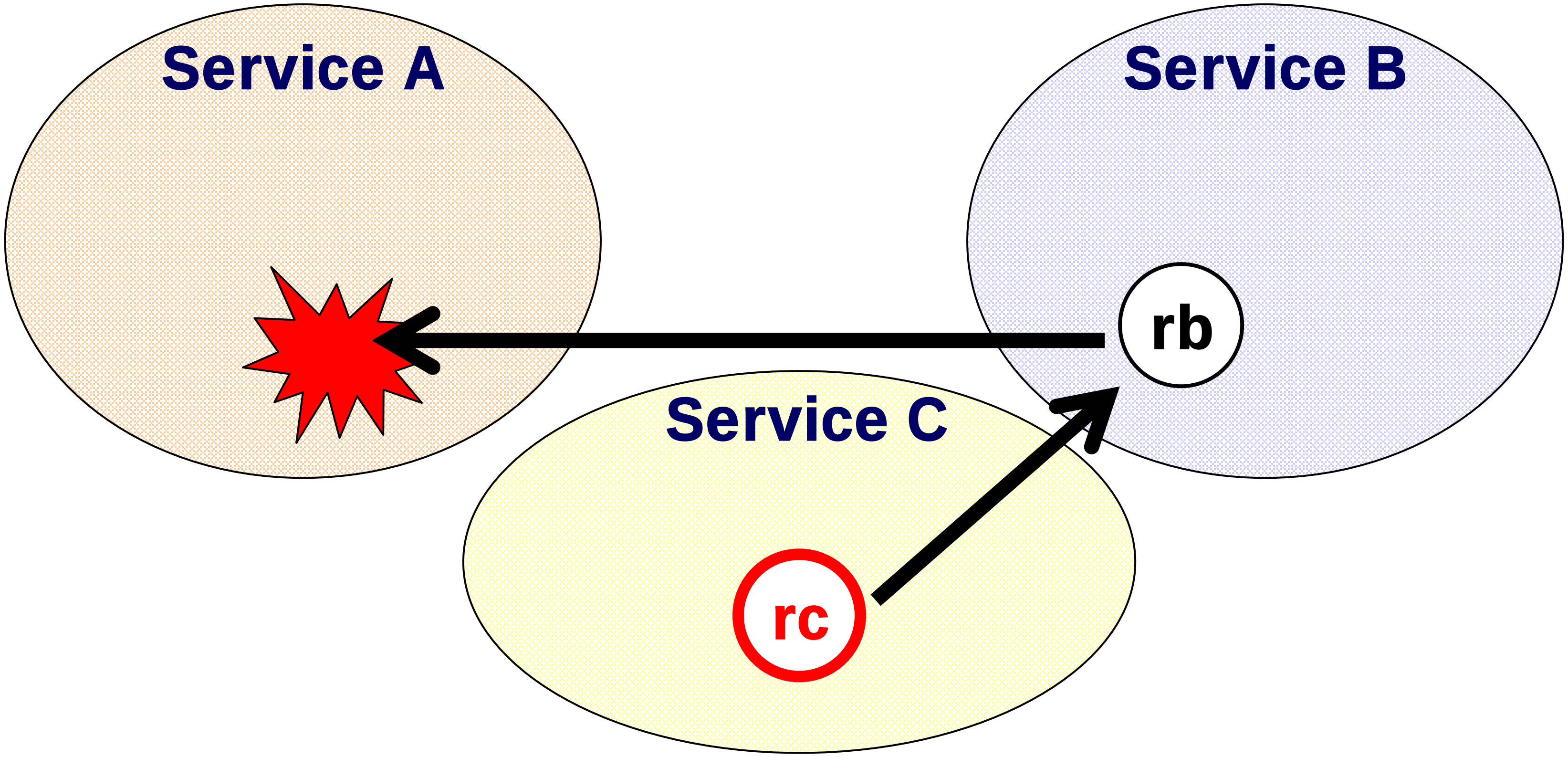
$\supseteq$ **the Pre-condition of rb2**

the Post-condition of ra

$\not\supseteq$ **the Pre-condition of rb**

Step4 Selection of rc

Select rc which creates execution conditions for rb.

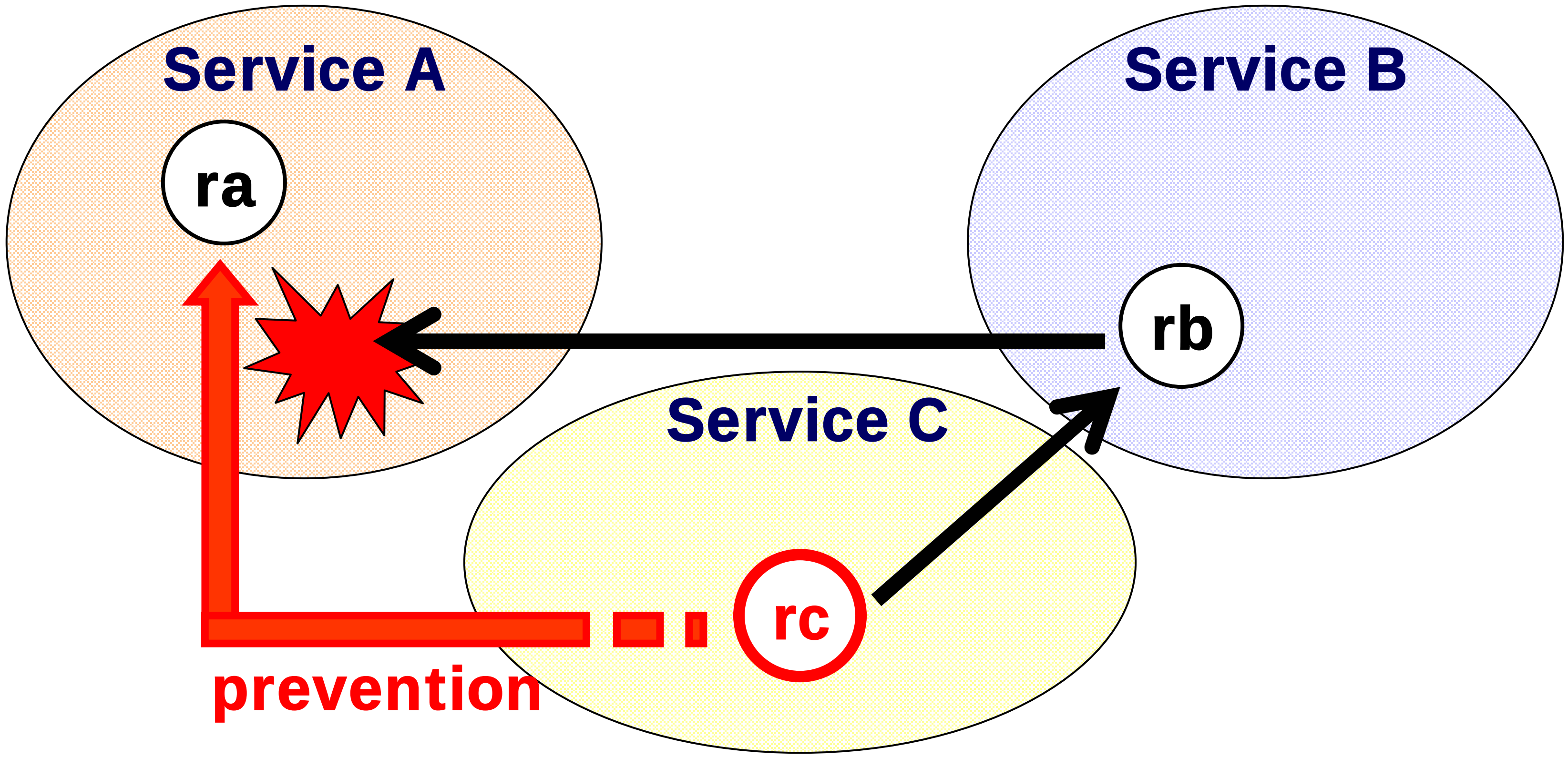


a) the Post-condition of rc

$\supseteq$ the Pre-condition of

Step4 Selection of rc

b) If selected rc has the same event as that of ra



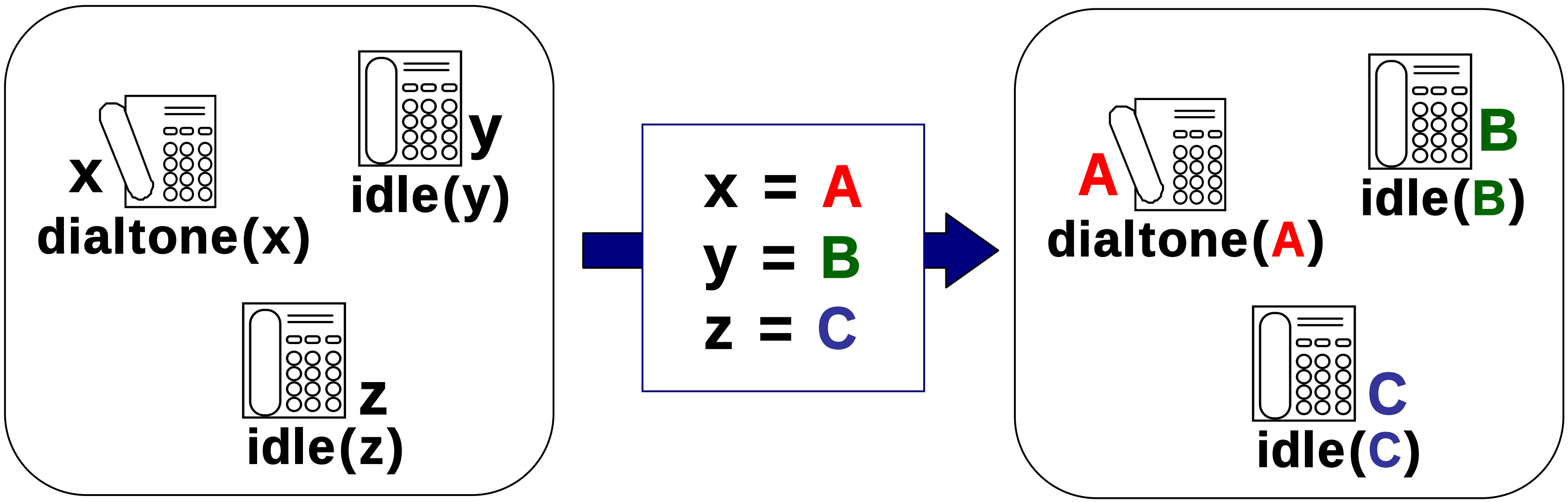
the Pre-condition of rc

$\supseteq$ the Pre-condition of ra

Experimental result

(c) \ (a) (b)	UPT			TCS			CFV			OCS			total
	CND	RC	TPC	CND	RC	TPC	CND	RC	TPC	CND	RC	TPC	
CFB	0	0	0	1	2	2	0	0	0	1	2	2	10
CND	-	0	0	-	0	0	-	0	0	-	0	0	0
RC	0	-	0	0	-	0	0	-	0	0	-	-	0
UPT	-	-	-	1	2	2	0	0	0	1	2	2	10
ACB	2	0	0	2	2	2	2	0	0	2	2	2	16
TCS	0	0	0	-	-	-	0	0	0	0	-	-	0
TWC	0	0	0	0	0	0	0	0	0	0	0	0	0
CFV	0	0	0	1	2	2	-	-	-	1	2	2	10
CW	2	0	0	2	1	1	2	0	0	2	1	1	12
TPC	0	0	-	0	0	-	0	0	-	0	0	0	0
ARC	3	0	0	3	3	3	3	0	0	3	3	3	24
OCS	0	0	0	0	0	0	0	0	0	-	0	0	0
total	7	0	0	10	12	12	7	0	0	10	12	12	82

Terminal assignment



Even for the same specification, depending upon terminal assignments, interactions occur or do not occur.

To detect all interactions, all ways of terminal assignments should be considered.

The general formula

the condition for selecting the rule :

set V		set U
pa(x), pa(y), ・・・	$\supseteq$	pa(A), pa(B), ・・・
the number of terminal variables : m (m ≧ n)		the number of real terminals : n

The number of ways for terminal assignments

$$_{m}P_{n}$$

The number of ways for terminal assignments

	the number of terminals
the constraint condition of service a	n1
rb	n2
rb2	n3
ra	n4
rc	n5

- step1 $_{n2}P_{n1}$
- step2 $_{n3}P_{n2}$
- step3 $_{n4}P_{n3}$
- step4 $_{n5}P_{n2}$

The number of terminal assignments to detect 3-way interactions

$$\therefore \quad \text{Max} \quad {}_{n2}P_{n1} \times {}_{n3}P_{n2} \times {}_{n4}P_{n3} \times {}_{n5}P_{n2}$$

- **Study1: 170 rules were investigated to obtain actually the number of primitives, that have the same primitive name and different arguments.**

number of the same primitive name	number of the rule	ratio
do not have	127	74.7%
two	35	20.6%
more than three	8	4.7%
total	170	100%

- **Study2: The number of terminal assignments for 28 combinations of four rules is investigated.**

the mean number of terminal assignments for all detected interactions : 1.3

It can be expected that the total number of terminal assignments is not actually a huge number.

Conclusion

♣ Conclusion

The mechanism for 3-way interactions occurrence and the detection algorithm based on the mechanism were proposed.

A detection system for 3-way interactions based on the proposed algorithm was implemented.

It was confirmed that the proposed detection algorithm is effective.

♣ Future work

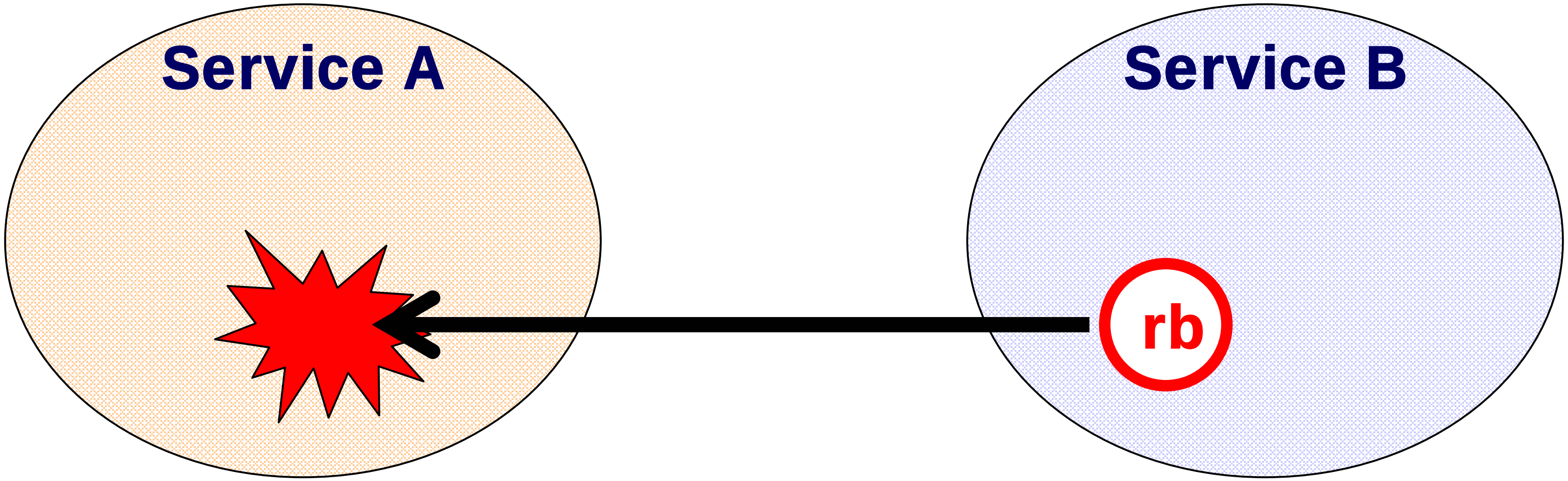
Other mechanisms should be investigated.

The algorithm to resolve the 3-way interactions detected should also be investigated.

Thank you for your kind attention!

Detection algorithm -Step1

the Post-condition of rb ⊇ states which contradict with specifications of service a



rb :

m-rc(y),achg(x,x,y) [achg(x,x,y)]: m-rc(y),achg(y,x,y)

"achg" means that it is to be charged.

Step1 Selection of rb

The condition for selecting a rule

the Post-condition of rb $\supseteq$

states which contradict with specifications of service a

the constraint of TCS	Post-condition of rb
achg(B,A,B)	m-rc(x) achg(x,y,x)

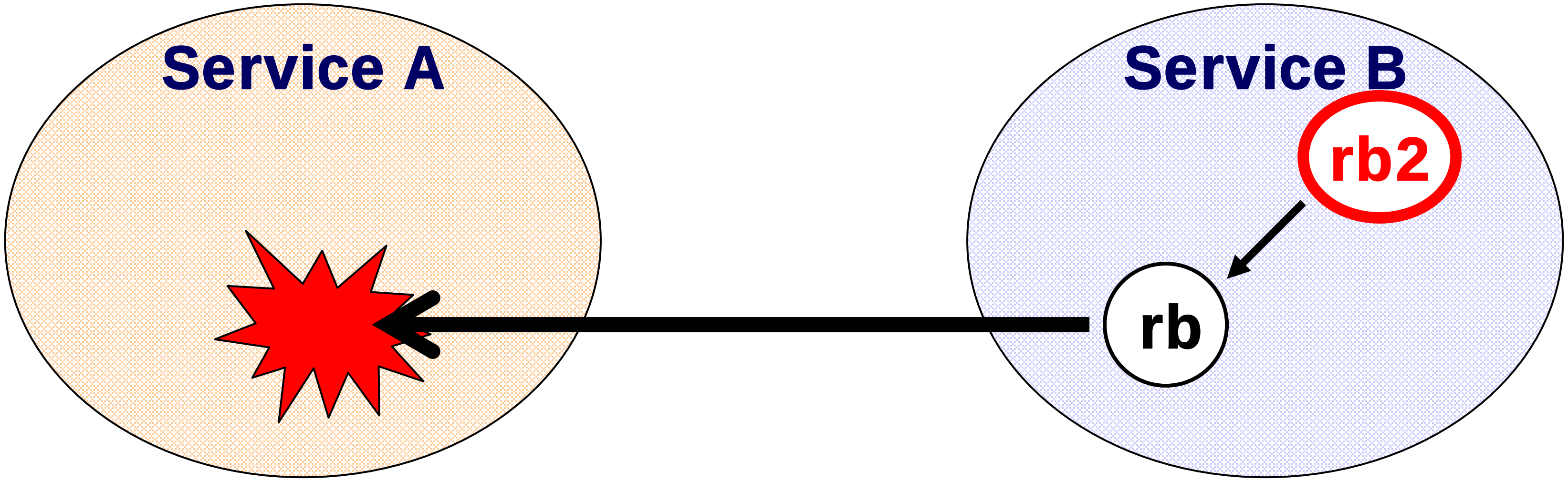
$$\begin{cases} x = B \\ y = A \end{cases}$$

m-rc(B)
achg(B,A,B)

the number of terminal assignments is one

Detection algorithm -Step2

the Post-condition of rb2
⊇ the Pre-condition of rb



rb2:
dialtone(x),idle(y) dial(x,y): Calling(x,y),achg(x,x,y)

Step2 Selection of rb2

The condition for selecting a rule

the Post-condition of rb2

$\supseteq$ the Pre-condition of rb

Pre-condition of rb	Pots-condition of rb2
achg(A,A,B)	Calling(x,y) achg(x,x,y)

$$\begin{cases} x = A \\ y = B \end{cases}$$

Calling(A,B)
achg(A,A,B)

the number of terminal assignments is one

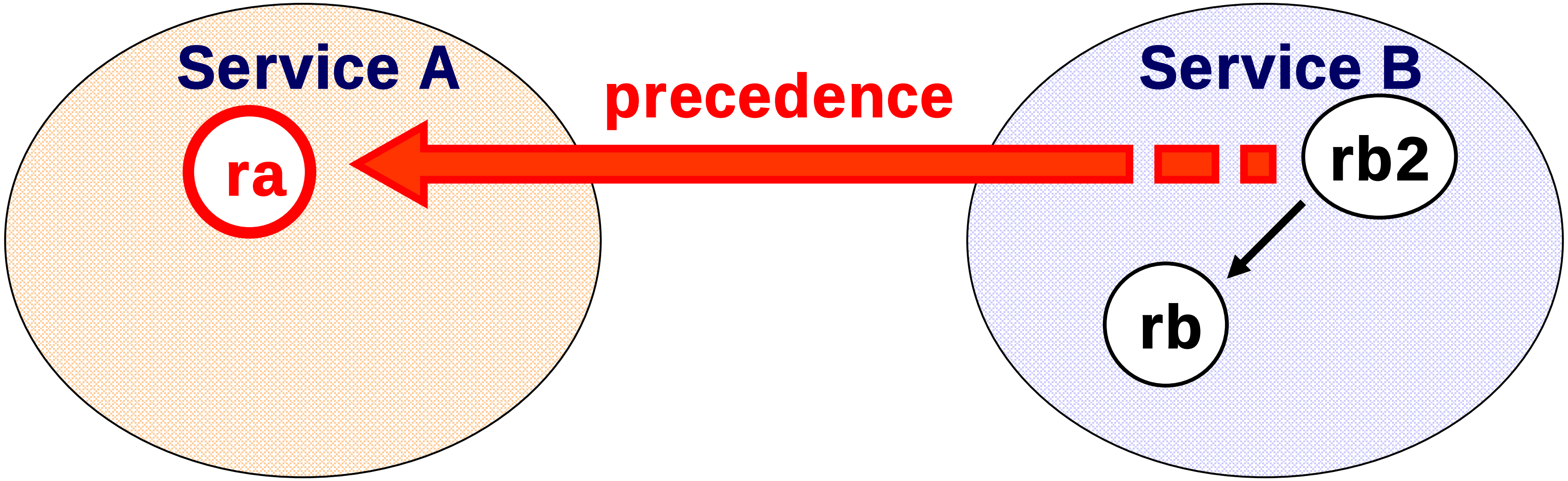
Detection algorithm -Step3

the Pre-condition of ra

$\supseteq$ the Pre-condition of rb2

the Post-condition of ra

$\not\supseteq$ the Pre-condition of rb



ra:

m-tcs(y,x),dialtone(x),idle(y) dial(x,y):

m-tcs(y,x),busy(x),idle(y)

Step3 Selection of ra

The condition for selecting a rule

the Pre-condition of ra $\supseteq$ the Pre-condition of rb2

Pre-condition of rb2	Pre-condition of ra
dialtone(A) idle(B)	dialtone(x) idle(y)

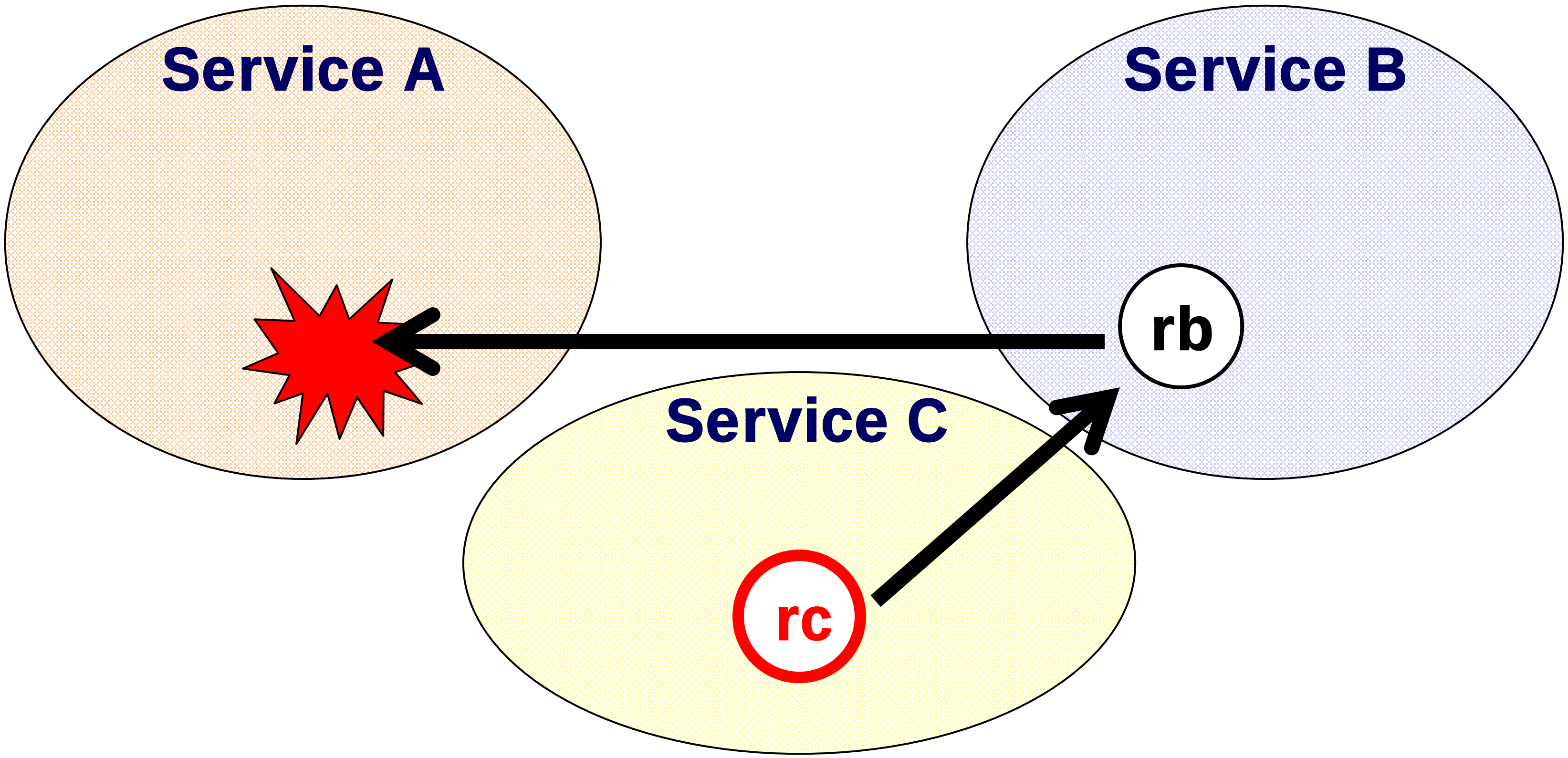
$$\begin{cases} x = A \\ y = B \end{cases}$$

dialtone(A)
idle(B)

the number of terminal assignments is one

Detection algorithm -Step4

the Post-condition of rc $\supseteq$ the Pre-condition of rb



rc:

m-cfv(y,z),dialtone(x),idle(y),idle(z) dial(x,y):
m-cfv(y,z),Calling(x,z),achg(x,x,y),achg(y,y,z)

Step4 Selection of rc

The condition for selecting a rule

the Post-condition of rc $\supseteq$ the Pre-condition of rb

Pre-condition of rb

achg(A,A,B)

Post-condition of rc

Calling(x,z)

achg(x,x,y) achg(y,y,z)

the number of terminal assignments is **two**

$\begin{cases} x = A \\ y = B \\ z = C \end{cases}$

$\begin{cases} x = C \\ y = A \\ z = B \end{cases}$

Calling(A,C)

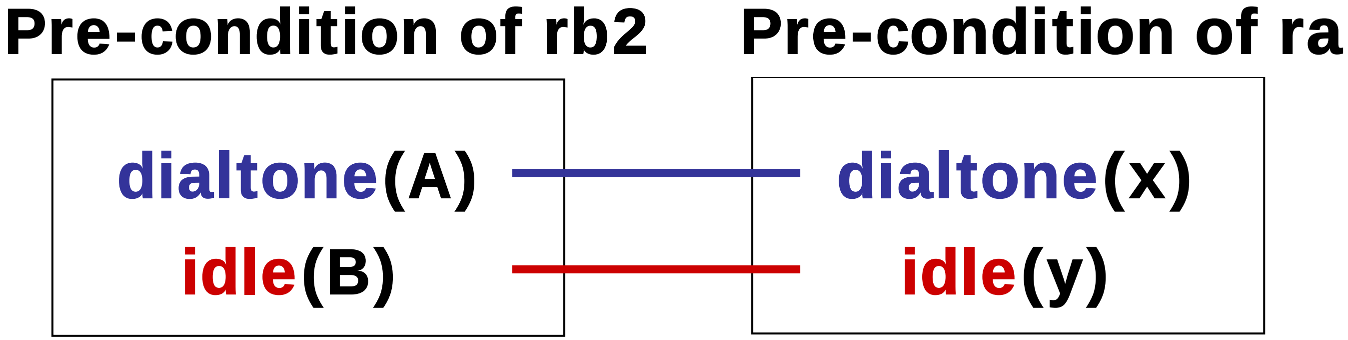
achg(A,A,B) achg(B,B,C)

Calling(C,B)

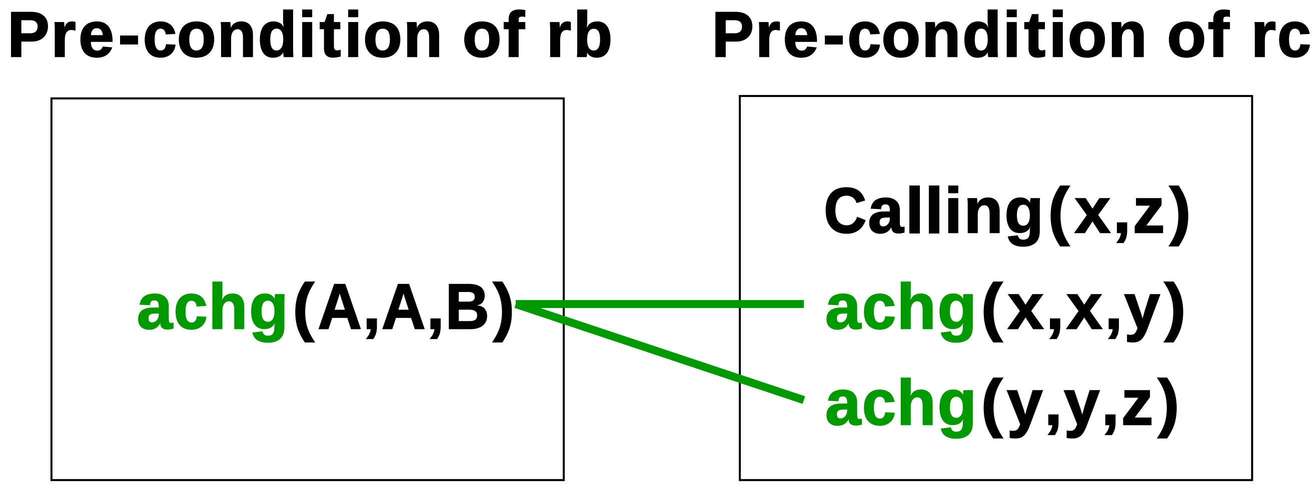
achg(C,C,A) achg(A,A,B)

Terminal assingment

Ex) Step3



Ex) Step4



In generally, there are more than one primitives which have the same primitive name.

Problem in implementing the detection system

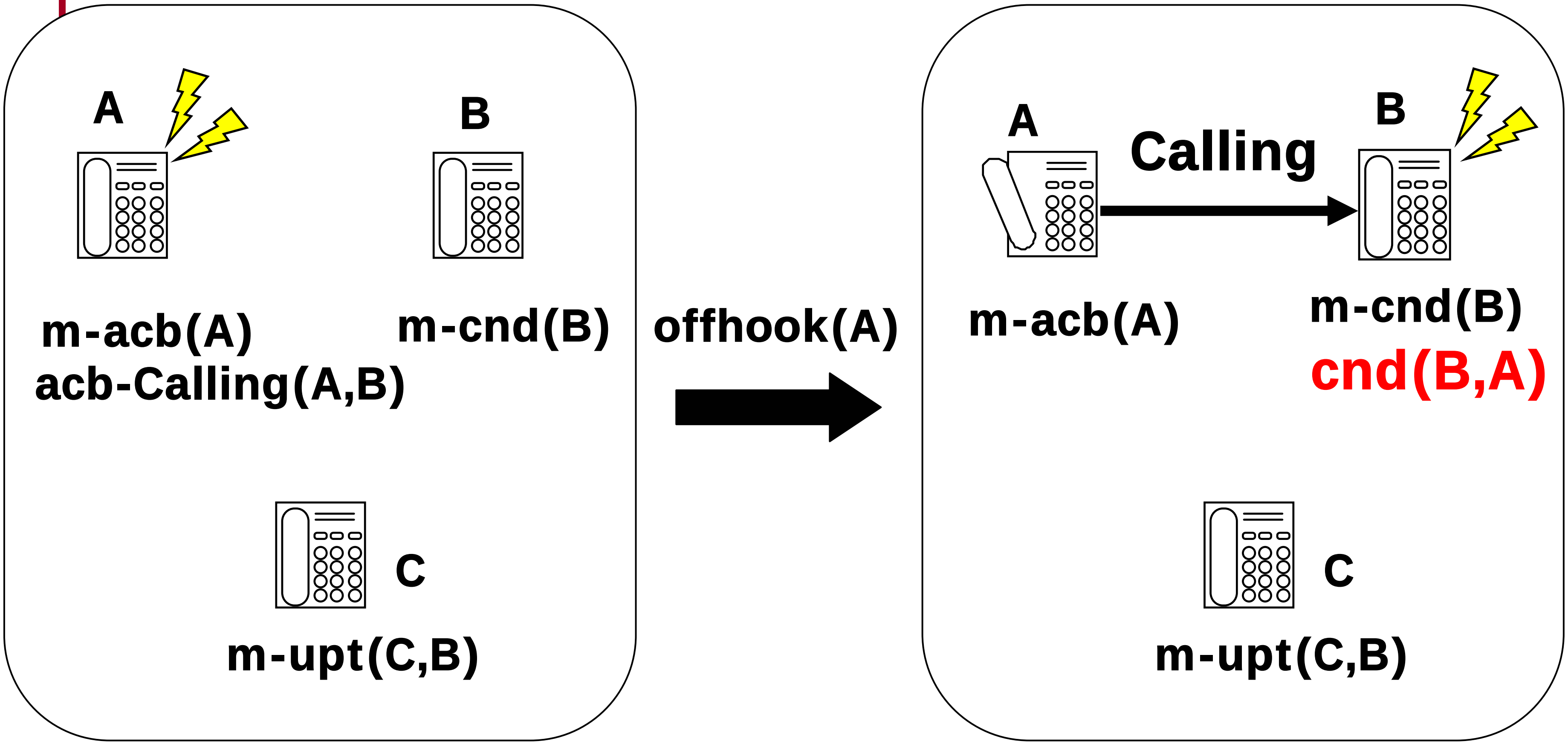
When the number of terminal variables in each rule is three, respectively, the maximum number of terminal assignment for detecting ...

- **2-way interactions** **33**
- **3-way interactions** **428,301**

This causes a possibility that the proposed detection algorithm cannot actually be used

We evaluated whether the detection algorithm can actually be used or not, from the view point of terminal assignment.

Example



UPT(Universal Personal Telecom.)

ACB(Automatic Call Back)

Services

- TCS(Terminating Call Screening)
- RC(Reverse Charge)
- CFV(Call Forwarding Variable)
- CFB(Call Forwarding Busy line)
- CND(Call Number Delivery)
- TWC(Three Way Call)
- UPT(Universal Personal Telecom.)
- ACB(Automatic Call Back)
- ARC(Automatic Re-Call)
- TPC(Third Party Charge)
- CW(Call Waiting)
- OCS(Originating Call Screening)